Using Manufacturing Message Specification for Monitor and Control at Venus

W. Randy Heuser, Richard L. Chen and Michael H. Stockett
Monitor & Control Technology Group
Jet Propulsion Laboratory
California Institute of ~'ethnology
4800 Oak Grove Drive
Pasadena, CA 91109

Abstract

The flexibility and robustness of a monitor and cent.ro] (M&C) system are a direct result of the underlying inter-processor communications architecture. A new architecture for M&C at. the Deep Space Communications Complexes (DSCCs) has been developed based on the Manufacturing Message Specification (MMS) process control standard of t-he Open System Inter connection (OS1) suite of protocols. This architecture has been tested both in a laboratory environment, and under operational conditions at. the Deep Space Network (DSN) experiment.a] Venus station (DS S- 1 3). The Venus experience in the application of OS1 standards to support. M&C has been ext. remely successful. MMS meets the functional needs of the station and provides a level of flexibility and responsiveness previously unknown in that. environment. The architecture is robust enough to meet Current operational needs and flexible enough to provide a migration pat h for new subsystems, This paper will describe the architecture of the Venus M&C system, discuss how MMS was used and the requirements this imposed on other parts of the system, and provide results from systems and operational testing at the Venus site.

Introduction

The Deep Space Network (DSN) operates the Venus facility at the Goldstone Deep **Space:** Communications Complex as a research site to conduct experiments in new tracking techniques, communication technologies and instrumentation. In response to expanding demand on station resources generated by the new **34-**meter Beam Waveguide Antenna, a new monitor and control system was developed and installed based on OSI protocols [1]. As an experimental facility, the Venus site is constantly changing. New instrumentation is moved to the station and tested on a weekly basis. The procedures to support observations or spacecraft activities change daily. However, Venus is also used to routinely for radio astronomy observations and the High Resolution Microwave Survey (HRMS). The Station Monitor and Control system was designed to support the constantly changing conditions at the station, while providing reliable support for operational Experiments.

Requirements

The basic requirements for Station Monitor and Control were derived from the requirements for monitor anti control of the DSN operational facilities. The operational facilities are a collection of computer automated antennas and support equipment which are linked together with a Local Area Network (LAN). The station is operated from a set of central control consoles which support the following requirements:

1) Allocation of station resources to provide assignment of subsystems in support of station activities.

2) Support data files are received at monitor and control and redistributed to subsystems as required.

3) Operator directives entered on monitor and control support the configuration and operation of subsystems.

4) Subsystem health and performance is displayed on monitor and control.

5) Subsystem event or alarm conditions aregenerated and reported to monitor and control.

6) Monitor data is exchanged between monitor and control and the active subsystems based on negotiated interface agreements.

In addition to these basic requirements, the constantly changing environment at Venus created the following requirements.

1) The Station Monitor and Control system must support the addition of a new subsystem without modification to the software.

2) The Station Monitor and Control system must support modification of displays and construction of new displays without new software deliveries.

The additional requirements reflect concerns for flexibility and low life cycle costs. The design of Station Monitor and Control makes extensive use of commercial software products with the tools that provide the required flexibility.

Station Monitor and Control Architecture

The OSI based monitor and control architecture enables operation of the Venus site through a single operator workstation.

OSI protocols are used to link the operator work station and the various station subsystems across the LAN. Commercial software packages based on the OSI Manufacturing Message Specification (MMS) protocol were employed to support all interprocessor communications. A commercial Supervisory Control and Data Acquisition (SCADA) software package was incorporated to support most of the operator inter-face functions through a graphical user interface (GUI). The extensive application of commercial software packages made it possible to build and install the system in one year at substantial cost savings [2].

There are two aspects to the OSI based monitor and central architecture. The first is the communications architecture which employs all seven layers specified in the OSI Basic Reference Model. Support for the communications architecture resides on all of the station computers. The second aspect involves the structure of the program (or programs) that run on each of the station computers. This structure is called the software architecture and is tightly coupled to the computer hardware, the operating system and the computer language employed. Since 98% of the software for Station Monitor and Control was purchased, much of the software architecture was set by the vendors. This discussion is therefore limited to those components used to integrate the commercial products into a working system. Both aspects of the architecture are depicted in Figure 1 and the station network configuration is seen in Figure 2.

Station Monitor and Control Implementation

The Station Monitor and Central system was initially implemented on an Intel 80386, 33 MHz computer running the Interactive UNIX operating system. The computer platform was provided to the project and selection of the software components was governed by the platform. The selection of the UNIX operating

4

systemwas made to provide a flexible multi-tasking environment . The platform and operating system directed the selection of the specific MMS/OSI software package . in addition, the platform anti operating system led to the selection of a Supervisory Control and Data Acquisition (SCADA) software package called Dexterity . The Dexterity product met all of the basic requirements to support the operator interface and provided an Application Program Interface (API) to support software development . The software development effort was limited to providing a link between Dexterity and the MMS network services .

The MMS software package provided a set of libraries and sample code to build MMS service calls . Using this package and the Dexterity API , the following elements were developed to support the MMS/OSI network services :

o 'l'he **MMS-Dexterity Network Server** provides all of the network services for Station Monitor and Control . The network server is a program that runs concurrent to Dexterity under the UNIX operating system and by UNIX convention is called a daemon .

o The **Dexterity Interface Program** is used to initiate the MMS-Dexterity network server to take action. The program is initiated to run when an operator uses the mouse or keyboard to trigger some network action .

o The **MMS-Dexterity** Timer **Program** runs concurrent to the: network server and provides a heartbeat to initiate polling of subsystem status data .

The key and most complex element of the software architecture is the MMS-Dexterity Network Server . The network server is started simultaneously with the Dexterity product anti is table

5

driven so changes to" the network can be accommodated without changing the network software. The network server establishes the environment to communicate across the LAN by reading a table with all of the network address information. Next, the network server reads a set of tables that identify all of the control and performance parameters used to operate the subsystems. Finally, the network server establishes communications with the Dexterity database.

Once initialized, the network server is ready to support all of the network communication services required to operate the complex from the Station Monitor and Control workstation. Station personal operate the complex through a graphical user interface (GUI) managed **by** the Dexterity software. The Dexterity software supports operator actions through the keyboard or a mouse. The Dexterity database services the user screens. In addition, the Dexterity product provides a set of tools to build and manage the GUI screens, The basic operation of the station is accomplished as follows:

1) Station resources are allocated to support Station activities by establishing a connection between Station Monitor and Control and the selected subsystems. Resources are released by concluding the connection. The process of establishing anti releasing a connection is accomplished using the mouse.

2) Subsystem health and perfor mance is read by, or reported to Station Monitor and Control using the MMS variable read service or the information report service. Changes in subsystem status anti performance are echoed on Station Monitor and Control using the same MMS services.

3) Operators _control_ a subsystem by entering Configurate ion
    parameters through the GUI i nto the Dexterity database .
    The operator action triggers the Dext erit y Interface
    Program to execut e and pass the select. ed parameters to
    the Dext eri ty Network Server .   The network server
    extracts the select ed data from the Dext eri ty database
    and writes the data across the net work to the subsystem.

4) Support data files are recei ved by Station Monitor and
    Control from JPL across the NASA Science Internet .
    Operators distribute support files to individual
    subsystems using the MMS file transfer services.   The
    opera t-or uses a De xt eri ty screen to enter the fi le name
    to be transferred, t_he destination file name and the
    dest. inat. ion subsystem.

5) Subsystem e vent or alarm con ditions are supported i n two
    ways .   Both methods combine the MMS information report.
    service with t-he Dext erity alarm service.   In the first.
    method, the subsystem i s programmed t o report data based
    on changing condit i ons .   The subsystem software engineer
    can report. on change, report on crit ical condit ion and/or
    report on return to nominal condition.   The Dexterity
    ciat.abase i s then conf i gured to alarm t he operator when a
    report ed dat a el ement goes outsi de i ts l imi ts .   in the
    second met. hod, the subsyst em i s programmed to report a
    text. message to Stat i on Monitor and Control .   The
    conditions that warrant a text. message are left to the
    discretion of the subsyst em software engineer.   The text
    message i s se nt to Stat ion Monitor and Control using the
    MMS informat ion report. service.   The Dext erit y dat.abase!
    i s conf igure d to di spl ay these text. messages as event s or
    alarms as a function of the message t ype ,

6) Monitor data is exchanged between directly between subsystems based on negotiated interface agreements using the MMS read variable service or the MMS information report service.

Subsystem Integration

All of the subsystems at the Venus site are based on the Intel 80x86 computer and the DOS operating system. Personal computers and the DOS operating system are frequently used by experimental subsystems because they are inexpensive and simple to program. DOS is a "single user" operating system and supports execution of only one program al_ a time. The subsystem program is structured to establish the subsystem configuration and status on execution, and to run in a continuous loop until an operator interruption triggers termination.

The key element in the application of MMS is the abstract model called the Virtual Manufacturing Device (VMD). The VMD model is used to describe t_he externally visible characteristics of a real device. Software modules which manipulate a real device are called VMD objects. These objects can be manipulated using MMS services such as context management, variable access, domain management, semaphore management. and event management services. State changes detected in the real device anti defined in the VMD model can trigger MMS services. For the implementation at Venus, a minimum of effort was required to make the subsys ems appear as VMD s.

To minimize the impact of the monitor and cent.rol effort on subsystem development, software routines were developed to provide network integration with a minimum of impact to the subsystem code. The impact to subsystem code is reflected in the pseudo-

code below :

```
main ( )
{
        dsn_init_mms () ;      /* routine  called to set up the
                                    network environment * /
        set_up_subsystem_environment ( ) ;
        while ( not_interrupted_by operator )
        {
               monitor_subsystem_operations () ;
               execute_subsystem_commands ( ) ;
               dsn_comm_server ( ) ;      /* routine called to check
                                    for  network  messages */
        }
        set_subsystem_in_safe_shut_down_conditions () ;
        dsn_terminate_mms ( ) ;   /* routine called to terminate
                                    the network environment */
        exit ();
}
```

A summary of the MMS services and routines developed for the
subsystems and the Station Monitor and Control workstation is
provided below.

MMS Context **Management   Services**

The routine *dsn_init_than* () is used by a client subsystem to
initiate communications with a *server*.

The routine *dsn_conclude* () is used by a Client subsystem to close
an association with a server subsystem.

The routine *dsn_listen* () is used by a server subsystem to open a

communication resource . The communication resources for each subsystem, commonly referred to as channels are established when the **subsystem** initiates operation . A subsystem that operates as a server will allocate listen channels for clients establish communications . The dsn listen () routine is provided to permit, a subsystem to allocate additional channels as server channels while operating, The *dsn stop listen ()* is used to close a channel in the listen mode .

The *dsn abort ()* is used by a client to terminate a client-server association without regard for pending request. s . This service is intended for use under special conditions and not to be used in place of the dsn conclude () service .

MMS File Management Services

The routine *dsn copy file ()* is *used by* a client subsystem to copy a file from a server subsystem.

The rout, ine *dsn send file()* is used by a client subsystemto invoke the MMS obtain file service in which a server subsystem copies a file from a client. subsystem.

The routine *dsn delete file ()* is used by a client subsystem to delete a file from a server subsystem.

MMS Variable Services

The routine *create dsn mms types ()* is used to est. ablish a variable as a network object . Subsystem implement.c)rs use this routine Lo register variables that will be managed across the network . The registration process identifies the address of the variable, its type anti size to the subsystem net-work software.

The routine *reg write var indic func()* is used to identify a
function to be called by the server, when a specific named
variable is written across the network , For example, a server
subsystem could have a physical switch that is set according to
the value of a named variable , By design, the switch must be
reset to its new condition on change of the named variable . The
server subsystem software engineer can write a function that
affects the physical change of the switch anti use the
reg write var indic func() routine to associate the function with
the named variable . When the write indication is received for the
associated named variable, the function is called.

The routine *reg_write varconfr func ()* is used to identify a
function to be called by the client, when a specific named
variable is written across the network and the write confirmation
is received by the client . Extending the example above, a client
could have a sequence of actions to perform i-hat, are dependent of'
the server subsystem switch position . The client subsystem
software engineer can write a function that affects action that.
can only occur after the switch reset. has been affected.   The
client function can be associated with the named variable write
confirmation using the reg write var confr func() rout. inc.   When
the write confirmation is received for the associated named
variable, the function is called .

The routine *reg read var indic func ()* is used to identify a
function to be called by the server, when a specific named
variable is read across the network .   For example, the status of
screw element of a server subsystem is reflected by a named
variable .   The subsystem software engineer can write a function to
update the status and associate that function with the specific
named variable using the reg read var indic func() routine.   When
the read indication is received, the function will be called and
the read response will reflect the latest status .

11

The *reg read var confr func ()* is used to identify a function to be called by tile client, when a specific named variable is read across the network and the confirmation is received by the client. Extending the example above, the client subsystem software engineer will write-e a function to take action on the status returned by the server subsystem. That function is associated with the specific named variable using the *reg_read_var_confr_func ()* routine. When the read confirmation is received by the client, the function is called,

The *reg info report. func ()* is used to identify a function to be called by t-he client, when a specific named variable is reported by a server using the *dsn send info rpt ()* service.

Summary

The Station Monitor and Cent.rc)1 system has been operating on a daily basis at the Venus site since December, 1992. The operation personnel have developed an extensive knowledge and understanding of the system. The stat.ion personnel perform routine configuration management., develop new tools and new displays as conditions changes.

The implementation of the Venus Station Monitor and Control has demonstrated that the MMS/OSI protocols can be used to support interprocessor communications in a spacecraft. tracking station. The integration of a commercial user interface package with an MMS network server has created a powerful, flexible tool to support the constantly changing requirements at. the site. Most of' the cost. savings derived from the MMS/OSI approach comes from the application of commercial products. The software developed in this activity to support the MMS network services on Station Monitor and Control is less that 20, 000 lines of code. The'

commercial software. products rc.present, more that. 750, 000 lines of code .   The functional capabilities provided through the combination of these two technologies offer great potential for cost. savings .

The integration of the two commercial products required three months with an additional three months of comprehensive testing be fore delivery to the station.   We have recently integrated a different SCADA product with the same MMS product. on a Sun SPARC. Though the API for the new SCADA product. was different, the process only required two months .   The table driven approach to the network interfaces and the SCADA products provide a flexible environment for station personal to respond to changing conditions.

The MMS/OSI communications architecture also provides a flexible environment to changing requirement.s.   The Virtual Manufacturing Device model for subsystems provides a single interface to network services.   When t-he High Resolution Microwave Survey project, discovered problems with their interface t_o the ant, tenna equipment, the implementation of a network gateway was a straight f orward process, requiring no modifications to the network .

The most. difficult part. of the effort. was upgrading the individual subsystems to enable them to operate over the LAN. Because the decision to proceed with an integrated, centralized M&C system was made after the subsystems began development. , the subsystems were, iri essence, ret.rof.it.t_ed to support. the new archite cture .   Fifty percent. of the Station Monitor & Cent.rc)l effort was spent. modifying the subsystems t_o include a LAN int erface, addressing problems whi ch occurred due to the limitations of the DOS operating system chosen by the subsystems, and int egrating a variety of commercial packages t o support

subsystem-specific probl ems .

The Venus Archit ecture is now serving as a baseline for testing aut omation concepts, The open, flexible architecture has greatly reduced integration time . Venus is serving as an operati onal test bed for the Li nk Moni tor & Cont rol Operator Assi st ant, an Art i f ici al Int el 1 i gence-based tool which aut oma tes conf igurat i on, cali bration, test, and operat i on of ground st at i on equipment [3] .

Acknowledgments

References

[1]   W. Randy Heuser, "An 0S1 Architecture for the Deep Space Network, " TDA Progress Report. 42-109, Jet. Propul si on Laboratory, Pasadena, CA, March 30, 1992.

[2]   W. Randy Heuser and Lynne P . Cooper, "An 0S1 Architecture for the DSN, " SpaceOps ' 9?., Pasadena, CA, November 1992.

[3]   Lorrine F. Lee and Lynne P. Cooper, "Operat i ons Automati on Using the Link Monitor & Cont rol Operator Assist ant, " t c) b e publ i shed i n the Proceedi ngs of Al AA Comput i ng i n Aerospace ' 9, San Diego, CA, Ott ober 1992
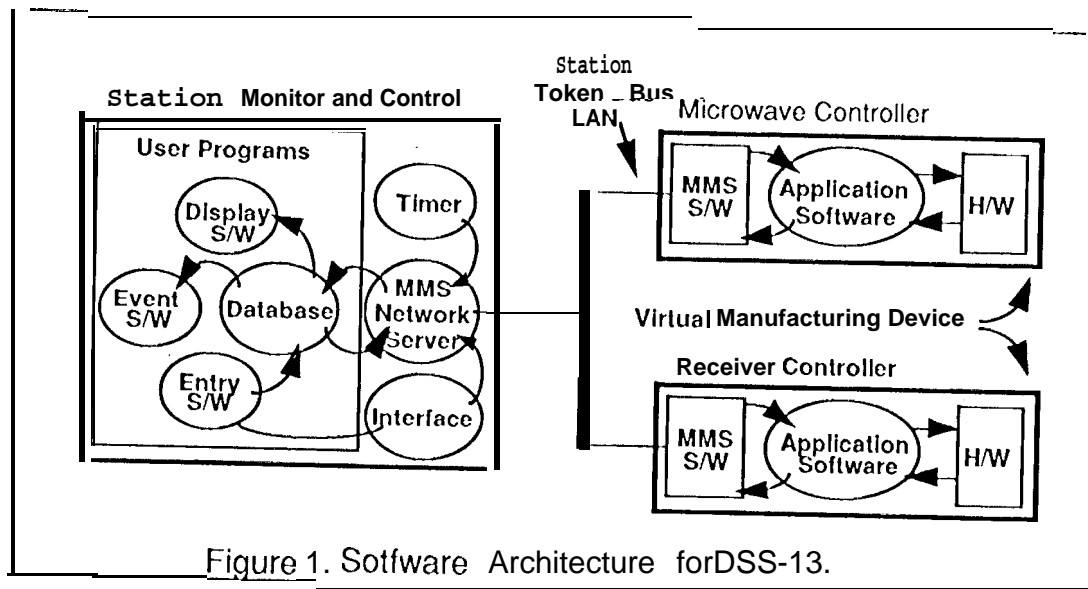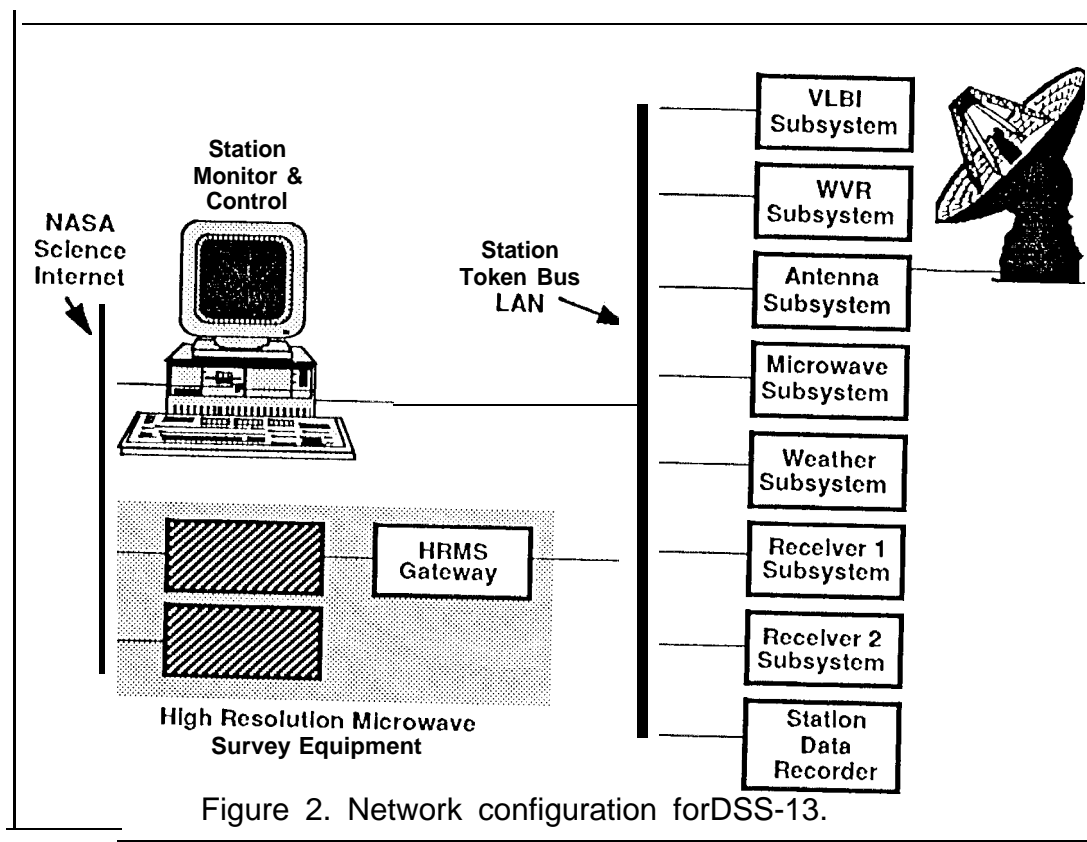
Figure 1. Sotfware Architecture forDSS-13.

Figure 2. Network configuration forDSS-13.